Most(?) Influential Papers in Program Synthesis

Bernd Fischer
ECS, University of Southampton, UK

What’s Program Synthesis?

“Automatic generation of executable programs from high-level (non-executable) specifications of their behaviour.” [Wikipedia]

What’s Program Synthesis?

“Automatic generation of executable programs from high-level (non-executable) specifications of their behaviour.” [Wikipedia]

Synthesis: Dreams → Programs

Z. Manna, R. Waldinger. IEEE Trans. Soft. Eng., 5(4), July 1979, 294-328.

What’s Program Synthesis?

“Automatic generation of executable programs from high-level (non-executable) specifications of their behaviour.” [Wikipedia]

Synthesis: Dreams → Programs

Z. Manna, R. Waldinger. IEEE Trans. Soft. Eng., 5(4), July 1979, 294-328.

Isn’t it all dreams???

What’s Program Synthesis?

“Automatic generation of executable programs from high-level (non-executable) specifications of their behaviour.” [Wikipedia]

Synthesis: Dreams → Programs

Z. Manna, R. Waldinger. IEEE Trans. Soft. Eng., 5(4), July 1979, 294-328.

Isn’t it all dreams???

No:

- Model-driven architecture
- Generative programming, templates, …
- Product-line development (?)
- …

Planware -- Domain-Specific Synthesis of High-Performance Schedulers

L. Blaine, L.-M. Gilham, J. Liu, D. Smith, S. Westfold
In: ASE 1998: 270-280

Influential?

- T. Emerson, M. H. Burstein: “Development of a Constraint-Based Airlift Scheduler by Program Synthesis from Formal Specifications.” (ASE 1999)
- K. E. Williamson, M. Healy: “Industrial Applications of Software Synthesis via Category Theory.” (ASE 1999 Best paper)
- D. Smith: “A Generative Approach to Aspect-Oriented Programming.” (GPCE 2004)
- D. Smith: “KIDS: A Semiautomatic Program Development System.” (IEEE Trans. Software Eng. 1990)

Planware -- Domain-Specific Synthesis of High-Performance Schedulers

L. Blaine, L.-M. Gilham, J. Liu, D. Smith, S. Westfold
In: ASE 1998: 270-280

immediate follow-up
(incl. Best paper)

Influential?

- T. Emerson, M. H. Burstein: “Development of a Constraint-Based Airlift Scheduler by Program Synthesis from Formal Specifications.” (ASE 1999)
- K. E. Williamson, M. Healy: “Industrial Applications of Software Synthesis via Category Theory.” (ASE 1999 Best paper)
- D. Smith: “A Generative Approach to Aspect-Oriented Programming.” (GPCE 2004)
- D. Smith: “KIDS: A Semiautomatic Program Development System.” (IEEE Trans. Software Eng. 1990)

Planware -- Domain-Specific Synthesis of High-Performance Schedulers

L. Blaine, L.-M. Gilham, J. Liu, D. Smith, S. Westfold
In: ASE 1998: 270-280

continuing development of underlying ideas

Influential?

- T. Emerson, M. H. Burstein: “Development of a Constraint-Based Airlift Scheduler by Program Synthesis from Formal Specifications.” (ASE 1999)
- K. E. Williamson, M. Healy: “Industrial Applications of Software Synthesis via Category Theory.” (ASE 1999 Best paper)
- D. Smith: “A Generative Approach to Aspect-Oriented Programming.” (GPCE 2004)
- D. Smith: “KIDS: A Semiautomatic Program Development System.” (IEEE Trans. Software Eng. 1990)

Planware

- Domain-specific program synthesis system
  “The goal of Planware is to allow experts in planning and scheduling to assembly quickly a specification of a scheduling problem, and to generate automatically a high-performance scheduler from it.”
- Built on top of Specware-system (evolved from KIDS)
- Uses formal (logic-based) specifications to capture design knowledge
  - data types *and* algorithms
- Uses category theory to structure specifications
  - morphisms, colimits, refinement
- Uses structure to guide program derivation process
- Uses theorem prover (automated+tactics) to derive program (correct-by-construction)

Why Influential?

- Synthesized programs…
  - … beyond toy level
  - … *orders of magnitude faster* than handcrafted scheduling code!
- Generic system core used by other people
- Supported by tools
- Heavy-duty theory put to work
  - not end in itself
  - well encapsulated (this is *Automated* SE!)
- One Small Step in a long-term research program

Knowledge-Based Synthesis of Numerical Programs for Simulation of Rigid-Body Systems in Physics-Based Animation

T. Ellman, R. Deak, J. Fotinatos. In: ASE 2002: 93-104

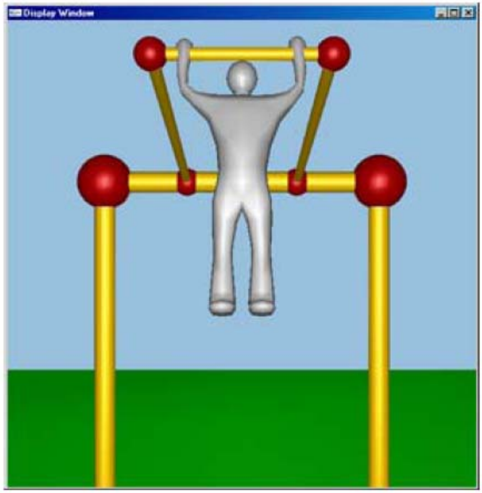
Influential?

- T. Ellman: Specification and Synthesis of Hybrid Automata for Physics-Based Animation. (ASE 2003 Best paper)
- B. Fischer, J. Schumann: Applying AutoBayes to the Analysis of Planetary Nebulae Images. (ASE 2003)
- T. Ellman, T. Murata: Deductive Synthesis of Numerical Simulation Programs from Networks of Algebraic and Ordinary Differential Equations. (KBSE 1996)

Ellman et al.

- Domain-specific program synthesis system

  "… a system for automatically synthesizing […] animation programs for a significant class of problems: constraint systems of rigid bodies, subject t driving and dissipative forces."



- Built on top of 3D Studio Max and Mathematica
- Use synthesis engine to capture design knowledge
  - hardcoded algorithm structure
  - specification language provides additional structure
- Use structure to guide program derivation process
- Use Mathematica to derive program
  - hardcoded ("algorithmically controlled") rewrite system

Why Influential?

- Built on top of commercial systems
  - leverage
  - simplicity
- Well-scoped: not trivial but not too big
  - feasible for small group